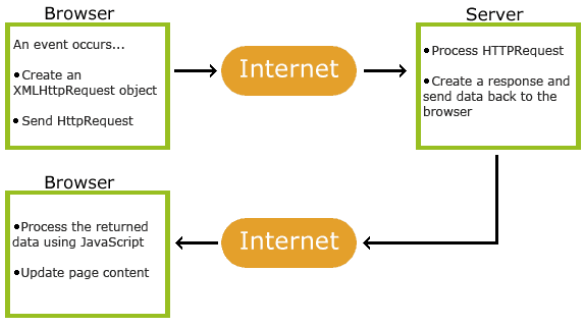
A AJAX INTRO

--> AJAX is a developer's dream, because you can:
1) Read data from a web server - after the page has loaded
2) **UPDATE A WEB PAGE WITHOUT RELOADING THE PAGE**
3) Send data to a web server - in the background

--> AJAX just uses a combination of:
1) A browser built-in XMLHttpRequest object (to request data from a web server)
2) JavaScript and HTML DOM (to display or use the data)

--> AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.



--> AJAX is a misleading name. AJAX applications might use XML to transport data, but it is equally common to transport data as plain text or JSON text.

--> AJAX can be used in json, xml, plain text, php, database, asp.

B AJAX THE XMLHTTPREQUEST OBJECT

--> Before that remember 3 rule :
1) Object in js is king
2) Object method syntax: `obj.method()`
3) Object properties syntax : `obj.propertyName` or `obj["propertyName"]`

--> The keystone of AJAX is the XMLHttpRequest object.

--> All modern browsers (Chrome, Firefox, IE7+, Edge, Safari, Opera) have a built-in XMLHttpRequest object.

--> The XMLHttpRequest object can be used to exchange data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

--> Create an XMLHttpRequest Object
1) Syntax : variable = new XMLHttpRequest();
//exp
```
Const xhttp =  new XMLHttpRequest();
```
2) Syntax for Older Browser : variable = new ActiveXObject("Microsoft.XMLHTTP");
3) To handle IE5 and IE6, check if the browser supports the XMLHttpRequest object, or else create an ActiveX object:
```
if (window.XMLHttpRequest) {
        // code for modern browsers
        xmlhttp = new XMLHttpRequest();
      } else {
        // code for old IE browsers
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
}
```

--> XMLHttpRequest Object Methods :

Method	Description
new XMLHttpRequest()	Creates a new XMLHttpRequest object ✔
abort()	Cancels the current request
getAllResponseHeaders()	Returns header information
getResponseHeader()	Returns specific header information
open(method, url, async, user, psw)	Specifies the request ✔ method: the request type GET or POST ✔ url: the file location ✔ async: true (asynchronous) or false (synchronous) ✔ user: optional user name psw: optional password
send()	Sends the request to the server ✔ Used for GET requests
send(string)	Sends the request to the server. ✔ Used for POST requests

setRequestHeader(header, value)	Adds HTTP headers to the request header: specifies the header name value: specifies the header value

--> XMLHttpRequest Object Properties :

Property	Description
onreadystatechange	Defines a function to be called when the readyState property changes ✔
readyState	Holds the status of the XMLHttpRequest. ✔ 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready ✔
responseText	Returns the response data as a string ✔
responseXML	Returns the response data as XML data
status	Returns the status-number of a request ✔ 200: "OK" ✔ 403: "Forbidden" 404: "Not Found" For a complete list go to the Http Messages Reference
statusText	Returns the status-text (e.g. "OK" or "Not Found")

--> Contoh simple :

1) di html :

```
<body>
  <div id="demo">  //isi dalam div ini kita ubah semua
  <h2>Belum berubah</h2>
  <button type="button" onclick="ubah()" name="button">Ubah Isi</button>
  </div>
</body>
<script type="text/javascript">
  ubah = () => {
    const xhttp = new XMLHttpRequest();        //bikin object xhttp
    xhttp.onreadystatechange = function() {    //ketika readyState mengalami perubahan
      if (this.readyState == 4 && this.status == 200) {      //4 berarti data sudah siap, 200 berarti OK
        document.getElementById("demo").innerHTML = this.responseText;   //responseText isi di text.txt
      }
    };
    xhttp.open("GET", "text.txt", true);      //request get buka text.txt asynchronously
    xhttp.send(); //kirim request
  }
</script>
```

2) isi text.txt : kata : data telah diubah

C PENJELASAN RINCI : DIBAGI 2 AJAX REQUEST DAN AJAX RESPONSE :

C 1 AJAX REQUEST

1) Send Request to A Server :

--> To send a request to a server, we use the open() and send() methods of the XMLHttpRequest object:

```
xhttp.open("GET", "text.txt", true);
            //get/post  //url     //asynchronous or synchronous
xhttp.send();
```

2) GET or POST ?

1) GET Request :

--> Simple GET request :

--> In the example above, you may get a cached result. To avoid this, add a unique ID to the URL:

```
xhttp.open("GET", "text.txt", true);
xhttp.send();
xhttp.open("GET", "text.txt?t="+ Math.random(), true);
xhttp.send();
```

--> If you want to send information with the GET method, add the information to the URL:

```
xhttp.open("GET", "text.txt?nama=hammam&lahir=1996", true);
xhttp.send();
```

2) POST Request:

--> Simple POST request :

```
xhttp.open("POST", "text.txt", true);
xhttp.send();
```

--> To POST data like an HTML form, add an HTTP header with setRequestHeader(). Specify the data you want to send in the send() method:

```
xhttp.open("POST", "text.txt", true);
```

```
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("nama=Hammam&lahir=1996");
```

3) URL - A File On a Server

--> The url parameter of the open() method, is an address to a file on a server:

```
xhttp.open("GET", "text.txt", true);
                                    //ini
```

--> The file can be any kind of file, like .txt and .xml, or server scripting files like .asp and .php (which can perform actions on the server before sending the response back).

4) Asynchronous - True or False?

--> Recommended : true

C 2 AJAX RESPONSE

1) The onreadystatechange Property

--> The readyState property holds the status of the XMLHttpRequest.

--> The onreadystatechange property defines a function to be executed when the readyState changes.

--> The status property and the statusText property holds the status of the XMLHttpRequest object.

--> The onreadystatechange function is called every time the readyState changes.

--> When readyState is 4 and status is 200, the response is ready

2) Using a Callback Function

--> A callback function is a function passed as a parameter to another function.

--> If you have more than one AJAX task in a website, you should create one function for executing the XMLHttpRequest object, and one callback function for each AJAX task.

--> The function call should contain the URL and what function to call when the response is ready :

```
loadDoc("url-1", myFunction1);
loadDoc("url-2", myFunction2);

function loadDoc(url, cFunction) {
 var xhttp;
 xhttp = new XMLHttpRequest();
 xhttp.onreadystatechange = function() {
   if (this.readyState == 4 && this.status == 200) {
    cFunction(this);
   }
 };
 xhttp.open("GET", url, true);
 xhttp.send();
}

function myFunction1(xhttp) {
  // action goes here
}
function myFunction2(xhttp) {
  // action goes here
 }
```

--> exp :

```
<div id="demo">
  <h2>The XMLHttpRequest Object</h2>
  <button type="button" onclick="jalankanAjax('ajax_info.txt', ubah)">Change Content
  </button>
</div>

<script>
  function jalankanAjax(url, cFunction) {
    var xhttp;
    xhttp=new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
      if (this.readyState == 4 && this.status == 200) {
       cFunction(this);
      }
    };
    xhttp.open("GET", url, true);
    xhttp.send();
  }
  function ubah(xhttp) {
  document.getElementById("demo").innerHTML =
  xhttp.responseText;
  }
</script>
```

3) Server Response Properties : responseText and responseXML

a) The responseText Property

--> The responseText property returns the server response as a JavaScript string, and you can use it accordingly:

```
document.getElementById("demo").innerHTML = xhttp.responseText;
```

b) responseXML : skipped

4) Server Response Methods : getResponseHeader() and getAllResponseHeaders(

a) The getAllResponseHeaders() method

--> returns all header information from the server response.

--> exp :

```
<p id="demo"></p>
<script>
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
      this.getAllResponseHeaders();
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
</script
```

--> Hasil :

date: Tue, 29 Jan 2019 23:04:29 GMT content-encoding: gzip vary: Accept-Encoding last-modified: Tue, 18 Jul 2017 16:14:29 GMT server: ECS (sgb/C7B3) x-powered-by: ASP.NET etag: "15bfdeee0ffd21:0+ident+gzip" x-frame-options: SAMEORIGIN x-cache: HIT content-type: text/plain status: 200 cache-control: public,max-age=14400,public content-length: 149

b) The getResponseHeader() method

--> returns specific header information from the server response.

--> exp :

```
<p>Last modified: <span id="demo"></span></p>
<script>
  var xhttp=new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
      this.getResponseHeader("Last-Modified");
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
</script>
```

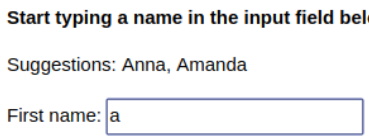
--> Hasil :

Last modified: Tue, 18 Jul 2017 16:14:29 GMT

D AJAX PHP

--> The following example demonstrates how a web page can communicate with a web server while a user types characters in an input field:

--> The function is triggered by the onkeyup event.



--> exp :

1) di .html :

```
<html>
<head>
<script>
   function showHint(str) {
     if (str.length == 0) {
       document.getElementById("txtHint").innerHTML = "";
       return;
     } else {
       var xmlhttp = new XMLHttpRequest();
       xmlhttp.onreadystatechange = function() {
       if (this.readyState == 4 && this.status == 200) {
         document.getElementById("txtHint").innerHTML = this.responseText;
       }
       };
      xmlhttp.open("GET", "gethint.php?q=" + str, true);
      xmlhttp.send();
     }
   }
</script>
</head>
<body>
   <p><b>Start typing a name in the input field below:</b></p>
   <form>
        First name: <input type="text" onkeyup="showHint(this.value)">
   </form>
   <p>Suggestions: <span id="txtHint"></span></p>
</body>
</html>
```

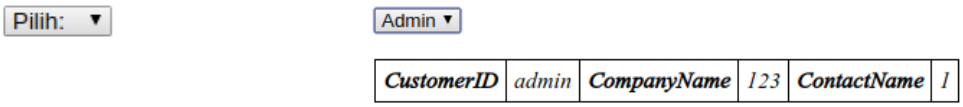
2) di php :

```
$a[] = "Raquel";
$a[] = "Cindy";
$a[] = "Doris";
```

```
$a[] = "Eve";
$a[] = "Evita";
$a[] = "Sunniva";
$a[] = "Tove";
$a[] = "Unni";
$a[] = "Violet";
$a[] = "Liza";
$a[] = "Elizabeth";
$a[] = "Ellen";
$a[] = "Wenche";
$a[] = "Vicky";

// get the q parameter from URL
$q = $_REQUEST["q"];

$hint = "";

// lookup all hints from array if $q is different from ""
if ($q !== "") {
  $q = strtolower($q);
  $len=strlen($q);
  foreach($a as $name) {
    if (stristr($q, substr($name, 0, $len))) {
      if ($hint === "") {
        $hint = $name;
      } else {
        $hint .= ", $name";
      }
    }
  }
}

// Output "no suggestion" if no hint was found or output correct values
echo $hint === "" ? "no suggestion" : $hint;
?>
```

--> di lempar ke .php pake url yg dimasukin this.value dari keyup, ditangkap $_REQUEST bisa juga pake $_GET, dilepar balik pake echo, ditangkap pake this.responseText

E AJAX DATABASE



--> Code :

1) di .html :

```
<body>
  <form action="">
    <select name="admin" onchange="showAdmin(this.value)">
      <option value="">Pilih:</option>
      <option value="admin">Admin</option>
    </select>
  </form>
  <br>
  <div id="txtHint">Hasil akan tampil disini</div>

  <script>
    function showAdmin(str) {
      var xhttp;
      if (str == "") {
        document.getElementById("txtHint").innerHTML = "";
        return;
      }
      xhttp = new XMLHttpRequest();
      xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
          document.getElementById("txtHint").innerHTML = this.responseText;
        }
      };
      xhttp.open("GET", "cobaAjaxPhp.php?q="+str, true);
      xhttp.send();
    }
  </script>
</body>
```

2) di .php :

```
<?php
$conn = mysqli_connect("localhost", "root", "", "pwd");
if($conn->connect_error) {
  exit('Could not connect');
}
```

```
$sql = "SELECT * FROM user WHERE username = '$_GET[q]';";
$hasil = mysqli_query($conn, $sql);
if ( mysqli_num_rows($hasil) > 0){
  while ($data = mysqli_fetch_assoc($hasil)) {
      echo "<table>";
      echo "<tr>";
      echo "<th>CustomerID</th>";
      echo "<td>" . $data['username'] . "</td>";
      echo "<th>CompanyName</th>";
      echo "<td>" . $data['password'] . "</td>";
      echo "<th>ContactName</th>";
      echo "<td>" . $data['role'] . "</td>";
      echo "</tr>";
      echo "</table>";
  }
}
?>
```

--> di lempar ke .php pake url yg dimasukin this.value dari select, ditangkap pake $_GET, dilepar balik pake echo, ditangkap pake this.responseText